Meeting	**HDB++**
Date	10-11.05.2016
Time/Place	Elettra
Contact	Lorenzo Pivetta lorenzo.pivetta@elettra.eu ++39 040 3758361
Partecipants	G.Abeille - Soleil R. Bourtembourg - ESRF R.Girardot - Soleil M.Ounsy - Soleil L.Pivetta - Elettra S.Rubio - Alba C.Scafuri - Elettra G.Scalamera - Elettra G.Strangolino - Elettra P. Verdier - ESRF L. Zambon - Elettra
Agenda	HDB++ meeting

Contents

1 **HDB++ status at Elettra/FERMI** 2

2 **HDB++ status at ESRF** 2

3 **HDB++ developments at ALBA** 2

4 **Extraction library/Qt** 3

5 **Egiga2m** 3

6 **Run/Shutdown management** 3

7 **Different polling periods for cache and event subsystems** 4

8 **Temporary storage of historical data** 4

9 **push_archive_event performance** 5

10 **HDB++ directory structure** 6

11 **GIT** 7

12 **TANGO meeting** 7

1 HDB++ status at Elettra/FERMI

HDB++ deployed on both FERMI (TANGO 9.2.2, 36 archiver instances managing 6544 Attributes) and Elettra (TANGO 8.1.2.c, 8 archiver instances managing 1269 Attributes. Both machines rely on MySQL database backends for history data. Still running the legacy HDB on both machines; for FERMI the complete switchover to HDB++ is forseen in the next 2 months. HDB++ archiving system integrated into the TANGO alarm system to alert of malfunctions. Moreover, e-mail alerts on alarms can be sent on a per-archiver basis with configurable recipients.

2 HDB++ status at ESRF

ESRF uses Cassandra as database backend for history data; 2 datacenters, 6 nodes, replication factor 3. HDB++ is currently archiving 7435 Attributes. No more MySQL schema in operation but available for testing. The old ESRF HDB (TACO) is still the official due to some servers auto-configutation needs. Working on analytics based on Apache Zeppelin and Spark.

3 HDB++ developments at ALBA

HDB++ deployed at the new MIRAS beamline, one archiver instance archiving 300 Attributes. PyTangoArchiving API anf Taurus GUI capable to configure and retrieve data from HDB++. ALBA pointed out pros and cons of HDB++; amongst the latter it is worth to mention:

- HDB++ *dedicated* archiving approach which is not suitable for dynamic data
- no temporary data storage support
- rules for table partitioning still to be defined
- not compatible with Mambo
- HDB++ schema larger disk usage

To mitigate the last point ALBA proposed to make recv_time and insert_time columns optional when creating the database. The archiver detects if the diagnostic timestamps are supported at runtime and possibly store them. Moreover, the error_desc can be possibly moved into a separate table. As an example, the existing and proposed table, in the case no diagnostic timestamps being configured, for a scalar read-only attribute are shown:

```
mysql> desc att_scalar_devboolean_ro;
+-------------+---------------------+------+-----+----------------------------+-------+
| Field       | Type                | Null | Key | Default                    | Extra |
+-------------+---------------------+------+-----+----------------------------+-------+
| att_conf_id | int(10) unsigned    | NO   | MUL | NULL                       |       |
| data_time   | timestamp(6)        | NO   |     | 0000-00-00 00:00:00.000000 |       |
| recv_time   | timestamp(6)        | NO   |     | 0000-00-00 00:00:00.000000 |       |
| insert_time | timestamp(6)        | NO   |     | 0000-00-00 00:00:00.000000 |       |
| value_r     | tinyint(1) unsigned | YES  |     | NULL                       |       |
| quality     | tinyint(1)          | YES  |     | NULL                       |       |
| error_desc  | varchar(255)        | YES  |     | NULL                       |       |
+-------------+---------------------+------+-----+----------------------------+-------+
```

```
mysql> desc att_scalar_devboolean_ro;
+-------------+---------------------+------+-----+----------------------------+-------+
| Field       | Type                | Null | Key | Default                    | Extra |
+-------------+---------------------+------+-----+----------------------------+-------+
| att_conf_id | int(10) unsigned    | NO   | MUL | NULL                       |       |
| data_time   | timestamp(6)        | NO   |     | 0000-00-00 00:00:00.000000 |       |
| value_r     | tinyint(1) unsigned | YES  |     | NULL                       |       |
| quality     | tinyint(1)          | YES  |     | NULL                       |       |
| error_id    | int(10) unsigned    | YES  |     | NULL                       |       |
+-------------+---------------------+------+-----+----------------------------+-------+
```

Moreover the additional error table contains the error_id column as primary key and the error_desc varchar, possibly indexed.

4 Extraction library/Qt

Data fetch configuration: fill from the past available with two mechanism: keep window or widen window. Support for time alignment of multiple data so that each set contains the same values on the x axis (time). Python binding available via SWIG. Support for fitting algorithms and/or data decimation under evaluation.

5 Egiga2m

New release of eGiga. Data is supposed to be organized as a set of unevenly spaced time series. Time series are taken from a web service which typically extracts data from a structured database. Drag-and-dop CSV file is also supported. Users can configure many plotting parameters such as plotting style, error visualization, number of Y axes.

6 Run/Shutdown management

Could be useful to support at least two (several?) operating modes, or strategies, in the archiving engine, such as Run, Shutdown, on a per-attribute basis (HDB++ meeting 08.04.2015). Currently the list of attributes for each archiver is stored in the property AttributeList, such as:

```
tango://srv-tango-srf.fcs.elettra.trieste.it:20000/eos/climate/18b20_eos.01/state
tango://srv-tango-srf.fcs.elettra.trieste.it:20000/eos/climate/18b20_eos.01/temperature
...
```

One requirement is to allow for a per-attribute configuration of contexts. The proposal is to modify the syntax of AttributeList device Property to support a *name=value* approach for the context; fields are separated by semicolon.

```
tango://srv-tango-srf.fcs.elettra.trieste.it:20000/eos/climate/18b20_eos.01/state; \
context=RUN|SHUTDOWN
tango://srv-tango-srf.fcs.elettra.trieste.it:20000/eos/climate/18b20_eos.01/temperature; \
context=RUN|SHUTDOWN
...
```

Keeping the current syntax for the attribute field allows for unchanged backwards compatibility. The labels for the strategy, implemented as enum, are defined in a free property, and/or in the class property and/or in the device property, with increasing priority. The defaults values, as well as the default context, are pre-defined but can be modified by the user. The default values are:

label	value
ALWAYS	0
RUN	1
SHUTDOWN	2
SERVICE	3

Whenever not specified the default context is ALWAYS. A new enum attribute, named **Context**, written by high layer logic, tells the archiver about the current context status or rather the required context transition.

Elettra will modify the EventSubscriber device server, ESRF the GUI.

7 Different polling periods for cache and event subsystems

The polling period affects several aspects of TANGO device server behaviour; relevant to clients and archiving system are:

- the rate at which the cache is updated
- the time at which change/archive/periodic event[s] are checked and possibly sent

Sometimes a *fast* rate for updating the cache is needed (short polling period) but on the opposite checking and sending change/archive/periodic events at that rate is unconvenient; event thresholds may help but not always solve.

One idea is to add *scale factors*, with default value equal 1, to define change/archive/periodic event check time with respect to the attribute polling period on a per-attribute basis; in pseudocode:

```
change_event_check_period = attribute_polling_period * change_event_polling_factor
archive_event_check_period = attribute_polling_period * archive_event_polling_factor
periodic_event_check_period = attribute_polling_period * periodic_event_polling_factor
```

8 Temporary storage of historical data

To support temporary storage of historical data one new column has to be added to the att_conf table:

```
CREATE TABLE IF NOT EXISTS att_conf(
att_conf_id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
att_name VARCHAR(255) UNIQUE NOT NULL,
att_conf_data_type_id INT UNSIGNED NOT NULL,
ttl INT UNSIGNED NULL DEFAULT 0,
facility VARCHAR(255) NOT NULL DEFAULT '',
domain VARCHAR(255) NOT NULL DEFAULT '',
family VARCHAR(255) NOT NULL DEFAULT '',
member VARCHAR(255) NOT NULL DEFAULT '',
```

```
name VARCHAR(255) NOT NULL DEFAULT '',
INDEX(att_conf_data_type_id)
) ENGINE=MyISAM COMMENT='Attribute Configuration Table';
```

The **ttl** column defines the time-to-live in hours on a per-attribute basis. Furtermore, to support configuration of this new field from the Configuration Manager interface should be updated:

```
int configure_Attr(
  string name,
  int type,               /* DEV_DOUBLE, DEV_STRING,... */
  int format,             /* SCALAR, SPECTRUM,... */
  int write_type,         /* READ, READ_WRITE,... */
  unsigned int ttl        /* hours, 0 = forever */
);
```

Deleting expired data is delegated to the SQL backend. Elettra will modify the EventSubscriber device server, ESRF the GUI.

9 push_archive_event performance

For the FERMI LowLevelRF subsystem acquisition at 50Hz repetition rate a real-time thread in a TANGO device server collects the data blocks coming from the hardware and fills 250 dynamic attributes. Events are pushed in the code.

Dynamic attributes	250
Attributes with push_archive_event by code	182
Attributes with archive threshold and/or period set	129

First case, check managed by TANGO core. Pseudocode:

```
set_archive_event(true, true);
...
while(1)
{
  //wait for data @ 50 Hz
  for(int i = 0; i < 182; i++) { //loop 182 attributes
    push_archive_event(att[i], ...);
  }
}
```

Second case, check managed in the server code. Pseudocode:

```
set_archive_event(true, false);
...
while(1)
{
  //wait for data @ 50 Hz
  for(int i = 0; i < 182; i++) { //loop 182 attributes
    if (archive_abs_thresh || archive_rel_thresh || archive_period) {
      push_archive_event(att[i], ...);
    }
  }
}
```

Additional tests using TANGO 9.2.2 are forseen.

	First case	Second case
push_archive_event calls / s	9100	~100
pushed archive events / s	~100	~100
CPU load	**First case**	**Second case**
events thread not running	~5%	~5%
no event subscribed	~100%	~8%
all events subscribed by 1 or 2 clients	~100%	~10%
push_event execution time	**First case**	**Second case**
no event subscribed	~20/900us	~130/210us
all events subscribed by 1 or 2 clients	~36/1000us **(losing events)**	~230/400us
full loop (182 attrs) execution time	**First case**	**Second case**
no event subscribed	~20ms	~0.45ms
all events subscribed by 1 or 2 clients	~24ms **(losing events)**	~0.65ms

Table 1: Performance figures, dual core PPC 1.3 GHz, TANGO 8.1.2.c

10 HDB++ directory structure

Current hierarchy:

```
doc
etc
gui
  |_ java-configurator
  |_ jhdbviewer
hdb++cm
hdb++es
hdbestractor
  |_ cpp
  |_ java
libhdb++
libhdb++cassandra
libhdb++mysql
libhdmmysql
```

Proposed hierarchy:

```
doc
gui
archiver
  |_ etc
  |_ hdb++cm
  |_ hdb++es
  |_ libhdb++
  |_ libhdb++cassandra
  |_ libhdb++mysql
  |_ libhdmmysql
hdbextractor
  |_ cpp
  |_ java
```

11 GIT

Elettra proposed to move HDB++ project to github. Since ESRF is working on TANGO migration to GIT we agreed to include the HDB++ project in the test patterns.

12 TANGO meeting

The next TANGO meeting agenda includes a presentation about HDB++ project status.